120A3003

Shubham Bhagwat
BE – IT
E1 Batch

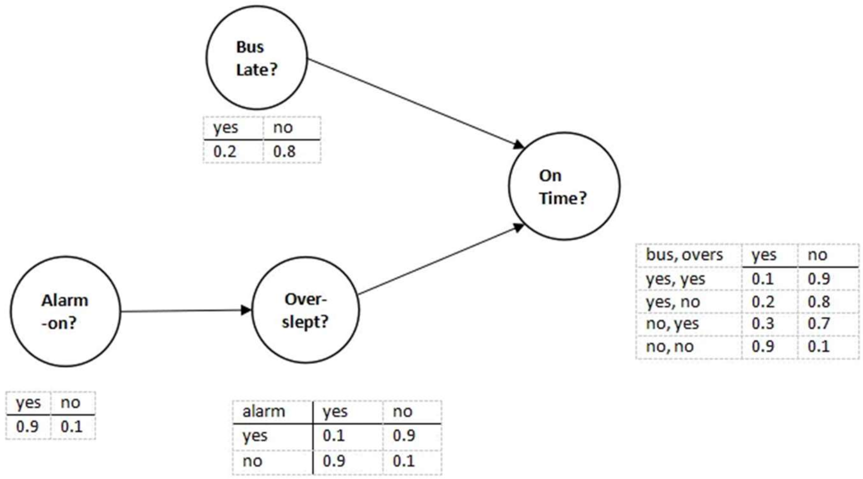
# Experiment No. 1

**Aim**: Implementation of Bayesian Belief Network

**Theory:**
A Bayesian network is a directed acyclic graph in which each edge corresponds to a conditional dependency, and each node corresponds to a unique random variable.
Bayesian network consists of two major parts: a directed acyclic graph and a set of conditional probability distributions
The directed acyclic graph is a set of random variables represented by nodes.
The conditional probability distribution of a node (random variable) is defined for every possible outcome of the preceding causal node(s).

Bayesian belief networks, nowadays are used in almost every field of machine learning, and artificial intelligence due to their less complex durability, and better approximation. This model is mostly used in those areas when a model is uncertain about the values of some event that has occurred at a specific area or a specific time. This helps the model to work in a competitive environment where the decision-making is on its own. From a technical point of view, there are different applications of Bayesian belief networks, some of the artificial intelligence, and machine learning fields that may have used these techniques are as; Image processing,

**Working of Bayesian Belief Network**
Bayesian networks' working is simple in nature. There are no complex variables or algorithms involved in the working of Bayesian belief networking unlike other machine learning or artificial intelligence models. These are simple graphical models having different edges and nodes. They have random variables available for working in the model, both dependent and independent relationships can be found between the variable using this technique. They can make models able to learn from the given data, they can become so strong after training and learning from the data that they can estimate the possibilities of some events. There are two main important parts to Bayesian belief networks, one is nodes which are basically the random variable in the tree or the data and the other one is the edge which represents the relationship between these nodes.




| yes | no |
|---|---|
| 0.2 | 0.8 |

| bus, overs | yes | no |
|---|---|---|
| yes, yes | 0.1 | 0.9 |
| yes, no | 0.2 | 0.8 |
| no, yes | 0.3 | 0.7 |
| no, no | 0.9 | 0.1 |

| yes | no |
|---|---|
| 0.9 | 0.1 |

| alarm | yes | no |
|---|---|---|
| yes | 0.1 | 0.9 |
| no | 0.9 | 0.1 |

**Dataset used**: Heart Disease Databases

The Cleveland database contains 76 attributes, but all published experiments refer to using a subset of 14 of them. In particular, the Cleveland database is the only one that has been used by ML researchers to this date. The “Heartdisease” field refers to the presence of heart disease in the patient. It is integer valued from 0 (no presence) to 4.

Attribute Information:

1. age: age in years
2. sex: sex (1 = male; 0 = female)
3. cp: chest pain type
4. Value 1: typical angina
5. Value 2: atypical angina
6. Value 3: non-anginal pain
7. Value 4: asymptomatic
8. trestbps: resting blood pressure (in mm Hg on admission to the hospital)
9. chol: serum cholestoral in mg/dl
10. fbs: (fasting blood sugar > 120 mg/dl) (1 = true; 0 = false)
11. restecg: resting electrocardiographic results
12. Value 0: normal
13. Value 1: having ST-T wave abnormality (T wave inversions and/or ST elevation or depression of > 0.05 mV)
14. Value 2: showing probable or definite left ventricular hypertrophy by Estes’ criteria
15. thalach: maximum heart rate achieved
16. exang: exercise induced angina (1 = yes; 0 = no)
17. oldpeak = ST depression induced by exercise relative to rest
18. slope: the slope of the peak exercise ST segment

19. Value 1: upsloping
20. Value 2: flat
21. Value 3: downsloping
22. thal: 3 = normal; 6 = fixed defect; 7 = reversable defect
23. Heartdisease: It is integer valued from 0 (no presence) to 4.

**Code:**

```
import numpy as np
import pandas as pd
import csv
from pgmpy.estimators import MaximumLikelihoodEstimator
from pgmpy.models import BayesianModel
from pgmpy.inference import VariableElimination

heartDisease = pd.read_csv('7-dataset.csv')
heartDisease = heartDisease.replace('?',np.nan)

print('Sample instances from the dataset are given below')
print(heartDisease.head())

print('\n Attributes and datatypes')
print(heartDisease.dtypes)

model=
BayesianModel([('age','heartdisease'),('gender','heartdisease'),('exang','heartdisease'),('cp','heartdisease'),('heartdisease','restecg'),('heartdisease','chol')])
print('\nLearning CPD using Maximum likelihood estimators')
model.fit(heartDisease,estimator=MaximumLikelihoodEstimator)

print('\n Inferencing with Bayesian Network:')
HeartDiseasetest_infer = VariableElimination(model)

print('\n 1. Probability of HeartDisease given evidence= restecg')
q1=HeartDiseasetest_infer.query(variables=['heartdisease'],evidence={'restecg':1})
print(q1)

print('\n 2. Probability of HeartDisease given evidence= cp ')
q2=HeartDiseasetest_infer.query(variables=['heartdisease'],evidence={'cp':2})
print(q2)
```

```
Sample instances from the dataset are given below
   age  gender  cp  trestbps  chol  fbs  restecg  thalach  exang  oldpeak  \
0   63       1   1       145   233    1        2      150      0      2.3
1   67       1   4       160   286    0        2      108      1      1.5
2   67       1   4       120   229    0        2      129      1      2.6
3   37       1   3       130   250    0        0      187      0      3.5
4   41       0   2       130   204    0        2      172      0      1.4

   slope ca thal  heartdisease
0      3  0    6             0
1      2  3    3             2
2      2  2    7             1
3      3  0    3             0
4      1  0    3             0

 Attributes and datatypes
age                int64
gender             int64
cp                 int64
trestbps           int64
chol               int64
fbs                int64
restecg            int64
thalach            int64
exang              int64
oldpeak          float64
slope              int64
ca                object
thal              object
heartdisease       int64
dtype: object
```

```
Learning CPD using Maximum likelihood estimators

 Inferencing with Bayesian Network:

 1. Probability of HeartDisease given evidence= restecg
+-----------------+---------------------+
| heartdisease    |   phi(heartdisease) |
+=================+=====================+
| heartdisease(0) |              0.1012 |
+-----------------+---------------------+
| heartdisease(1) |              0.0000 |
+-----------------+---------------------+
| heartdisease(2) |              0.2392 |
+-----------------+---------------------+
| heartdisease(3) |              0.2015 |
+-----------------+---------------------+
| heartdisease(4) |              0.4581 |
+-----------------+---------------------+

 2. Probability of HeartDisease given evidence= cp
+-----------------+---------------------+
| heartdisease    |   phi(heartdisease) |
+=================+=====================+
| heartdisease(0) |              0.3610 |
+-----------------+---------------------+
| heartdisease(1) |              0.2159 |
+-----------------+---------------------+
| heartdisease(2) |              0.1373 |
+-----------------+---------------------+
| heartdisease(3) |              0.1537 |
+-----------------+---------------------+
| heartdisease(4) |              0.1321 |
+-----------------+---------------------+
/usr/local/lib/python3.10/dist-packages/pgmpy/models/BayesianModel.py:8: FutureWarning: BayesianM
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/pgmpy/models/BayesianModel.py:8: FutureWarning: BayesianM
  warnings.warn(
```

**Conclusion:**

Implemented Bayesian Belief Network in Python successfully.